Python Fun(damentals)

Initializing your packaging system

Alexander Rymdeko-Harvey

Obscurity Labs

```
* Directory Structure
* Config Files
```

Copyright (c) 2020 Obscurity Labs LLC.

Using `poetry` to start a project

`poetry` provides a handy set of tools to set up a project.

This command will help you kickstart your new Python project by creating a directory structure suitable for most projects:

```
$ poetry new lab-project
Created package lab_project in lab-project
$ tree
.
├── completed
├── lab-project
│   ├── lab_project
│   │   └── __init__.py
│   ├── pyproject.toml
│   ├── README.rst
│   └── tests
│       ├── __init__.py
│       └── test_lab_project.py
├── README.md
└── slides
    ├── 02_01_scaffolding_your_project.md
    ├── 02_02_init_packaging_system.md
    └── 02_03_down_slecting_packages.md
```

Copyright (c) 2020 Obscurity Labs LLC.

Using `poetry` to update your decencies

First, if we pinned a version we need to update it from exact to `greater` then version notation, this tells `poetry` that we want the newest version above `2.0.0` as that's the minimal tested requirement.

```
[tool.poetry]
name = "lab-project"
version = "0.1.0"
description = ""
authors = ["Your Name <you@example.com>"]

[tool.poetry.dependencies]
python = "^3.7"
requests = "2.0.0" ---> "^2.0.0"

[tool.poetry.dev-dependencies]
pytest = "^5.2"

[build-system]
requires = ["poetry>=0.12"]
build-backend = "poetry.masonry.api"
```

Copyright (c) 2020 Obscurity Labs LLC.

Using `poetry` to update your decencies cont.

After we have updated our pinned package we can simply update our packages which will `lock` them for future builds. This provides a strong guarantee that the code running on these decencies will work:

```
$ poetry update
Updating dependencies
Resolving dependencies... (0.1s)

Writing lock file

Package operations: 0 installs, 0 updates, 5 removals

  - Removing certifi (2020.4.5.1)
  - Removing chardet (3.0.4)
  - Removing idna (2.9)
  - Removing requests (2.23.0)
  - Removing urllib3 (1.25.9)
```

Copyright (c) 2020 Obscurity Labs LLC.

Lab_1.py

Tasking

Using the new `poetry` command perform the following:

1. Go into the `02_python3_packaging_managment/lab-project` folder, notice we provided a full test project already.
2. Using `poetry` install run `poetry install` and then `poetry shell`
3. Update the `pyproject.toml` requests package to `^2.0.0` and run a update.

Testing your work

Now we updated our packages we want to always make sure our unit tests pass!

```
$ poetry shell
$ pytest tests/test_lab_project.py
```

Copyright (c) 2020 Obscurity Labs LLC.